

Remarks

The present amendment responds to the Official Action dated November 17, 2003. The Official Action rejected claims 1-6 under 35 U.S.C. §102(a) based on Smith U.S. Patent No. 5,878,224 ("Smith"). Claims 7 and 8 were rejected under 35 U.S.C. §103(a) based on Smith in view of Yoshimura et al. U.S. Patent No. 6,125,397 ("Yoshimura"). These grounds of rejection are addressed below following a brief discussion of the present invention to provide context.

Claims 1-8 have been amended to address antecedent basis issues found in preparation of this amendment. Claims 5 and 7 have also been amended to be placed in proper form for storage medium and carrier wave claims. Claims 1, 3, 5, and 7 also have been amended to add either a queuing step or a queuing means to clarify what happens to the datagram when the prescribed threshold is not exceeded. Dependent claims 9-14 have been added to cover certain aspects of the present invention. Claims 1-14 are presently pending.

The Present Invention

The present invention recognizes that the consequences of intentional datagram flooding attacks and unintentional overload situations resulting from a burst of connectionless datagrams can be mitigated by dropping the traditional notion of attempting to distinguish between legitimate and illegitimate traffic. In the present invention, both legitimate and illegitimate datagram traffic is subject to a common policy

that attempts to guarantee that legitimate work will be performed and a server will not crash in flooding situations, irrespective of whether the flooding is caused by legitimate or illegitimate datagram traffic. The present invention helps to prevent a server from crashing due to overload and it prevents one or more attackers from consuming all resources on a network server.

According to the present invention, in response to the arrival of a datagram destined for a specified port on a network server, the transmitting host is identified from the datagram and the number of datagrams already queued for the same host and for the same port is determined. If this number exceeds a prescribed threshold, the datagram is queued in a queue slot of the port.

The prescribed threshold is dynamically determined in the preferred embodiment. The owner of the network server specifies for each port that is subject to datagram flooding checks a maximum number of queued datagrams (M) allowed at any time to the port and a controlling percentage (P) of available queue slots remaining for the port. The present invention keeps track of the number (A) of queued datagrams for the port and it calculates the number of available queue slots (I) by subtracting the number of queued datagrams from the maximum number of datagrams ($I = M - A$). If the number of datagrams already queued for the transmitting host is equal to or greater than P times the number of queue slots left ($M \geq P \cdot I$), then the present datagram is not queued for the port. Otherwise, the datagram is queued and the number of queued datagrams (A) for the port is incremented by one.

The Art Rejections

All of the art rejections hinge on the application of either Smith standing alone or a combination of Smith and Yoshimura. As addressed in greater detail below, relied upon art does not support the Official Action's reading of it and the rejections based thereupon should be reconsidered and withdrawn. Further, the Applicant does not acquiesce in the analysis Smith and Yoshimura made by the Official Action and respectfully traverses the Official Action's analysis underlying its rejections.

Claims 1-6 were rejected under 35 U.S.C. §102(b) based on Smith. Smith is entitled "System For Preventing Server Overload By Adaptively Modifying Gap Interval That Is Used By Source To Limit Number Of Transactions Transmitted By Source To Server." It describes an approach which addresses various connection based telecommunication networks. Smith, col. 3, lines 49-59. Turning to Fig. 2 of Smith, Smith's approach addresses telecommunication networks utilizing connection based protocols which require control signaling connections, shown as broken lines in Fig. 2, and Asynchronous Transfer Mode (ATM) connections, shown as solid lines in Fig. 2. The control signaling connections are used to allocate logical paths over an ATM network to establish connections for subsequent transport of video or voice information. Smith, col. 3, line 60-col. 4, line 6. Smith's approach simply addresses preventing server overload in that connection based environment.

To prevent server overload, Smith's approach includes a controller which establishes a target incoming transaction workload per measurement interval. During the measurement interval, a server computes an admission factor representing the fraction of new transaction requests a source may send to a server. The server communicates the admission factor to a source in response to new transactions from the source. If the controller allows a source to send the first message of a transaction over an established connection, it also must allow any network node to send subsequent messages corresponding to that transaction. See, Smith, col. 5, lines 30-40. As a result, Smith's approach requires consideration of past messages allowed in its determination of whether to allow present messages to be sent.

In contrast, the present invention addresses defending against network flooding attacks of connectionless datagrams. In response to the arrival of a datagram from a host for a port on a network server, the number of datagrams already queued to the port from the host is determined. If the number of datagrams already queued to the port from the host exceeds a prescribed threshold, the datagram is discarded. Otherwise, the datagram is queued to the port. Claim 1, as presently amended, reads as follows:

1. A method of preventing a flooding attack on a network server in which a large number of connectionless datagrams are received for queuing to a port on the network server, comprising:
 - connectionless datagram from a host for a port on the network server, if the number of connectionless datagrams already queued to the port from the host exceeds a prescribed threshold;
 - discarding the datagram, if the number of connectionless datagram already queued to the port from the host exceeds the prescribed threshold; and

queuing the connectionless datagram to a queue slot of the port, if the number of connectionless datagram already queued to the port from the host does not exceed the prescribed threshold.

Smith does not teach and does not suggest preventing connectionless datagrams from flooding a network server. More particularly, Smith does not teach and does not suggest "determining, in response to the arrival of a connectionless datagram from a host for a port on the network server, if the number of connectionless datagrams already queued to the port from the host exceeds a prescribed threshold" as presently claimed. See also claims 3, 5, and 7.

The Official Action suggests that the text of Smith at col. 1, lines 35-38 discloses a large number of connectionless datagrams. Applicants respectfully disagree. The cited text which is found in Smith's Background of the Invention section addresses a problems of hot to meet service demands which are growing in a volatile manner. At col. 1, lines 22-34, Smith describes examples of services as voice activated dialing, local number portability, calling name delivery and other screening features, automated telephone polling, and personal communication services.

Typically, to establish these services, a connection protocol is necessary to establish the underlying session connection. Smith, col. 1, lines 30-33 and col. 3, lines 49-59. One of ordinary skill in the art would recognize that connectionless datagrams are classified under the category of User Datagram Protocol (UDP) of an Internet Protocol stack. The protocols used in Smith are connection protocols which are classified under the category of Transmission Control Protocol (TCP) which require an

established connection before transmitting meaningful data. See, for example, Douglas Comer, *Internetworking with TCP/IP Principles, Protocols, and Architecture*, p. 137, Prentice Hall 1988. A copy of this page is enclosed herewith for ease of reference as Exhibit A hereto. Thus, it is sent that Smith's approach addresses an entirely different approach within a separate and distinct protocol class context. Unlike a connectionless approach as taught in the present invention, once a first message of a transaction is allowed, any subsequent message corresponding to that transaction must also be allowed. See Smith, col. 5, lines 38-40. The present invention which utilizes a connectionless protocol need not consider subsequent messages related to a transaction.

The Official Action further suggests that the disclosure of Smith at col. 5, lines 29-33 stands for "a prescribed threshold." In particular, the Official Action language quotes the following language: "the output of a server overload controller is a computed value." Applicants respectfully disagree. In the cited text, the computed value represents "the fraction of new transaction requests a source may send to the server during the coming measurement interval." In contrast, the present invention utilizes the prescribed threshold "in response to the arrival of a connectionless datagram". The number of datagrams already queued to the port from the host is compared against the prescribed threshold to determine if the received datagram will be queued to a port or dropped.

Claims 7 and 8 were rejected under 35 U.S.C. §103(a) based on Smith in view of Yoshimura. The fallings of Smith are not cured by the teachings of Yoshimura.

Yoshimura addresses overall network congestion and thus considers the amount of network resources consumed throughout the network in response to congestion conditions. To this end, Yoshimura's approach monitors an amount of the network used by data transfers of computers connected with each other by the network. See, for example, Yoshimura, col. 21, line 56 – col. 22, line 17. Yoshimura's approach further allows a recovery-type congestion control mechanism and an avoidance-type congestion control mechanism to coexist on the network. Yoshimura, col. 3, lines 56-58. Coexistence means that a congestion state can be controlled when the congestion avoidance-type data transfer and the congestion recovery-type data transfer are performed through a transmission path and share a network resource on that transmission path. Col. 3, lines 60-63.

Unlike Yoshimura, claims 7 and 8 address a carrier wave containing program code to determine congestion at a network server by a particular transmitting host. The program code, when activated on the network server, determines if the number of datagrams already queued to a port on the network server from the transmitting host exceeds a prescribed threshold. If so, the activated program code discards the datagram. Otherwise, the program code allows the datagram to queue to the port. Claim 7, as presently amended, reads as follows:

7. A carrier wave containing program code that is operable by a network server for preventing a flooding attack on the network server in which a large number of datagrams are received for queuing to a port on the server, the program code including instructions for causing the network server to execute the steps of:

determining, in response to receipt of a datagram from the host for queuing to the port on the network server, if the number of datagrams already queued to the port from a host exceeds a prescribed threshold; discarding the datagram, if the number of datagrams already queued to the port from the host exceeds the prescribed threshold; and queueing the datagram to the port, if the number of datagrams already queued to the port from the host does not exceed the prescribed threshold.

Yoshimura and Smith, either separately or in combination, do not teach and do not suggest a carrier wave containing program code having instructions to execute the steps of “determining, ... , if the number of datagrams already queued to the port from a host exceeds a prescribed threshold, in response to a datagram from the host for queuing to the port on the network server” as presently claimed. Further, Yoshimura and Smith, either separately or in combination, do not teach and do not suggest the instruction steps of “discarding the datagram, if the number of datagrams already queued to the port from the host exceeds the prescribed threshold” or “queueing the datagram to the port, if the number of datagrams already queued to the port from the host does not exceed the prescribed threshold” as presently claimed.

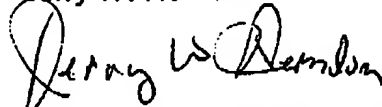
The Official Action apparently only relies on Yoshimura as standing for a “carrier wave containing program code” and cites Yoshimura, col. 21, line 56 – col. 22, line 20. Although Yoshimura discloses a claim in a conventional carrier wave claim format, the steps carried out by a computer in Yoshimura do not teach and do not suggest the “determining”, “discarding”, and “queueing” steps as claimed.

The relied upon references fail to recognize and address the problem of preventing a flooding attack by datagrams in the manner advantageously addressed by the present claims. The claims as presently amended are not taught, are not inherent, and are not obvious in light of the art relied upon.

Conclusion

All of the presently pending claims, as amended, appearing to define over the applied references, withdrawal of the present rejection and prompt allowance are requested.

Jerry W. Herndon



Attorney Representing Applicant
Reg. No. 27,901

IBM Corporation
Intellectual Property Law
Department T81/Building 503
P.O. Box 12195
Research Triangle Park, NC 27709

(919) 543-3754

FAX: 919-254-4330
Email: herndonj@us.ibm.com

Library of Congress Cataloging-in-Publication Data

Coxe, Douglas.
Interworking with TCP/IP : principles, protocols, and
architecture / Douglas Coxe.

p. cm.
Bibliography: p.
Includes index.

ISBN 0-13-470154-2

1. Computer networks. 2. Computer network protocols. 3. Data
transmission systems. I. Title.
TK5105.5.C39 1988 87-35201
004.6—dc19 CIP

Editorial/production supervision: Ellen B. Greenberg
Cover illustration: Jim Kinstry
Cover design: Bruce Kossel
Manufacturing buyer: Cindy Grant

UNIX is a registered trademark of AT&T Bell Laboratories.
PROBT-10 is a trademark of Proteon Corporation.
VAX, Microvax, and LSI 11 are trademarks of Digital Equipment Corporation



© 1988 by Prentice-Hall, Inc.
A Division of Simon & Schuster
Englewood Cliffs, New Jersey 07632

All rights reserved. No part of this book may be reproduced, in any form
or by any means, without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6 5

ISBN 0-13-470154-2 025

Prentice-Hall International (UK) Limited, London
Prentice-Hall of Australia Pty. Limited, Sydney
Prentice-Hall Canada Inc., Toronto
Prentice-Hall Hispanoamericana, S.A., Mexico
Prentice-Hall of India Private Limited, New Delhi
Prentice-Hall of Japan, Inc., Tokyo
Simon & Schuster Asia Pte. Ltd., Singapore
Editora Prentice-Hall do Brasil, Ltda., Rio de Janeiro

EXHIBIT A
S/N 09/503,608
RSW-00-0010

Transmission Control Protocol

understand the principle of sliding windows, we can examine the service provided by the Internet. The service is defined by the *Transmission Control Protocol* or *TCP*. The reliable stream service is so important that the entire protocol suite is referred to as *TCP/IP*. It is important to understand that:

PL is a communication protocol, not a piece of software.

between a protocol and the software that implements it is analogous between the definition of a programming language and a compiler. The lines are blurred. People encounter TCP software much more frequently than they do TCP specifications, so it is natural to think of a particular implementation as the standard. Nevertheless, the reader should try to distinguish between

Does the TCP protocol provide? TCP is complex, so there is no simple protocol specifies the format of the data and acknowledgements that need to achieve a reliable transfer, as well as the procedures the computer must follow if it the data arrives correctly. It specifies how TCP software distinguishes multiple destinations on a given machine, and how communications can occur between machines without errors like lost or duplicated packets. The protocol also specifies how to maintain a TCP stream transfer and how they agree when it is complete. It doesn't attempt to understand what the protocol does not include. Although it describes how application programs use TCP in general terms, it does not specify the details of the interface between an application program and TCP. That is left to the programmer. This section discusses the operations TCP supplies, but it does not specify the arguments to those procedures that implement the operations. It specifies the interface between the application program and TCP. That is left to the programmer. This section discusses the operations TCP supplies, but it does not specify the arguments to those procedures that implement the operations. It specifies the interface between the application program and TCP. That is left to the programmer.

provided assumes little about the underlying communication system with a variety of packet delivery systems including the Internet service. For example, TCP can be implemented to use dialup telephone network, a high speed fiber optic network, or a lower speed long-haul network. The large variety of delivery systems TCP can use is one of its

12.7 TCP Ports And Connections

Like the User Datagram Protocol (UDP) presented in Chapter 1, TCP resides above IP in the Internet protocol layering scheme. Figure 12.5 shows the conceptual organization.

Conceptual Layering

Reliable Stream (TCP)	User Datagram (UDP)
Internet (IP)	
Network Interface	

Figure 12.5 The conceptual layering of UDP and TCP above IP. TCP provides a reliable stream service, while UDP provides an unreliable datagram delivery service. Application programs access both.

TCP allows multiple application programs on a given machine to communicate concurrently and it demultiplexes incoming TCP traffic among application programs. Like the User Datagram Protocol, TCP incorporates abstract objects called *ports* that identify the ultimate destination within a machine. Each port is assigned a small integer used to identify it. Because port number assignments are local to a given machine and unique within that machine, the ultimate destination for TCP traffic is uniquely specified by giving both a destination host Internet address and a port number.

Unlike UDP, TCP is a connection oriented protocol that needs two endpoints to make communication meaningful. Before TCP traffic can pass across the Internet, application programs at both ends of the connection must agree that the connection is desired. To do so, the application program on one end performs a *passive open* function by contacting its operating system and indicating that it will accept an incoming connection. At that time, the operating system assigns a port number for one end of the connection. An application program at the other end must then contact its operating system. The *active open* request to establish a connection. The two TCP modules using an *acknowledge* that the connection is established. Once a connection has been created, the TCP software modules at each end can begin passing data.